**Anomaly description**:   
*[static, see next page for dynamic]*

To describe an anomaly, we have multiple features of the dataset and how that will constitute a part of the anomaly. For each feature, I will choose an acceptable threshold/range of values and assume all other values to be part of the anomaly. Additionally, I will mention if there are other features upon which it depends to be considered an anomaly:

1. **Timestamp** - There are only certain hours of the day during which any of the operations can be performed on a bucket by a user. The other factors might depend on the timestamp, but I assume that the acceptable threshold is:  
     
   [09:00 - 05:00] on all days of the week, assuming people might work on all days of the week and might want to perform some operations.  
     
   Possible anomalies -   
   - Outside authorized hours access
2. **User** - The threshold for this is an access list of users [a,b,c] allowed to access these buckets at acceptable time frames.   
     
   Possible anomalies -   
   - User accessing the buckets at non-regular times  
   - Unauthorized user from access list accessing the server  
   - *User accessing outside their usual pattern of access(based on timeframes, resources and actions they usually access)*
3. **Action** - There are only a list of actions to be performed, and this factor cannot be a part of the anomaly
4. **Resource ID** - There is a list of buckets that are allowed to be accessed [x,y,z] and only these resources are allowed to be accessed. No other buckets can be accessed.

Possible anomalies -

* Other resources apart from the access list are accessed
* *Based on the usual pattern of access(timestamp and user with specific actions to these resources), something is out of pattern*

**Dynamic Analysis**

From the above, it is quite clear that defining a static threshold might not help identify other outlier factors for the patterns arising from the dataset’s distribution of data.

I’m using Isolation forest first over here due to the smaller size of the dataset and easier thresholding of random samples.

So I’m using the **Isolation forest** with the following threshold details in order to isolate the anomaly points in the dataset. There’s 2 important ways to determining the reason behind why a record is an anomaly and their conclusions:

* **Thresholding**: Find the records which have decision score < threshold(anomalies) and determine the pattern of which features have the highest impact
  + Based on the input data and a contamination factor of 0.2, it seems that the threshold value is 0. That is, any value below 0 is considered an anomaly.
  + Another important observation from the output\_anomalies.csv file, is we can draw conclusions of the reason based on the decision scores and observe the recurring values of the feature(based on its feature value) to draw a conclusion of the features that are causing it to be anomalies
* **Shapley values**: Shap\_values for the dataset features [if the value is closer to 1 and positive]
  + Based on the summary plot for the visualization, we get that **Timestamp** is the highest impact feature to determine if a record is an anomaly or not.
  + The impact of each feature follows the order - Timestamp > User > Action > Resource
  + Identifying the records with the values close to the Timestamps in the range of the output.csv helps identify the exact threshold values for the individual features based on the SHAP values for each of the features
  + Thus, the reason behind why a record is an anomaly depends on a number of features’ Shapley values and are in the columns why\_anamoly\_<column\_name>

Based on this, the possible anomalies are based on the **threshold value of the decision\_scores** and **higher** **Shapley values** of the features from the IsolationForest model to help determine why those records are anomalous. And the reason is, because the values are outside the forest of decision trees that are constructed from the input dataset.

This is how I would describe an anomaly.

Other techniques for detecting anomalies using unsupervised learning - [to compare outputs based on hyperparameter tuning]

* Local Outlier Factor
* Robust Covariance
* One-class Support Vector Machine

For each of these models, the rest of the determination algorithm would remain the same